

CONSUMER ELECTRONIC SHOW, that semi-annual get-together, is two weeks away. Dick Hauser and I will be at Las Vegas to see all the new and marvelous things the Electronics Industries have for us to buy. We'll be at the AstroVision booth to get acquainted and re-acquainted with dealers, distributors, buyers, etc., and the AstroVision team. Full report next time.

INCENTIVE While we'll be talking to top level people in the distributing chain, it may be a while before the information trickles down to the individual retailer and his sales personnel. So I'd like to enlist your support in spearheading an attack on the working troops - make them aware of: 1) the ARCADIAN; 2) the existence of the Basic cartridge and what it can do; and 3) the existence of our advertisers and their products. Accordingly, we are making up a little package of this data and would like you to give copies to any retailers in your area. This data will be coded, and when the retailer contacts us, we'll send you a buck for your trouble. The package will be completed when we get back from Las Vegas.

NEW USER GROUP is forming up in the Eastern part of Cleveland. Contact Jack Walters, 630 Broadway Ave., Bedford, OH 44146 (439-2542) for details as to meeting dates, etc. The place will be his store (Visual Aides), which will be a fully stocked Arcade dealer, including third-party programs on tape from our advertisers (and possibly others - are there any?).

CONTEST STORY: There was no contest last month because there were no entries. The story is a little different this time - lack of time. I had to rush this into print because the printer was taking the week off for the holidays, and there wasn't enough time to get the tapes to the judges. The contest will be back next month - but entries are always needed. And we need 'regular' programs as well.

#### MACHINE CODE

The advent of the Bit Fiddler's cartridge (check the ad on p.26 and the review) now allows us to bypass the Basic cartridge entirely and to communicate directly with the Z-80 microprocessor. Previously, we had to utilize some of the hidden (unpublished) features of the Basic cartridge in order to access the Z-80 to use machine code. The first of a number of articles about our success in so doing will be found in Volume 1, page 25. Subsequent articles and programs brought us to the point in the last Volume where we could generate characters, and where the computer was doing most of the work of entering the code.

Now by replacing the Basic cartridge with the Bit Fiddler's product, we no longer have to use "work-around" techniques.

The Z-80 microprocessor operates by means of a set of instructions that tell it what to do in a highly defined way. These directions are supplied by the chip manufacturer in a document called the "Z-80 Instruction Set". We would also need to understand the principles of machine code operation in order to utilize the instructions properly. We plan to provide some introductory material in these pages for those interested in this subject.



## TUTORIAL      MAGIC REGISTER by Brett Bilbrey

The Bally Arcade has a very powerful graphics register that not very many people know about, or know how to use. Let's find out what it is, where it is, what it does, and mostly, how we can use it. But first its name: MAGIC REGISTER

The Magic Register is a hardware register (storage space) that exists in the custom chips. It is port addressable, which means it is accessed by the  $\&(X)=Y$  construct from Basic, or the OUT command from machine language. It is classified as an output port which means that values may only be sent to it, never received from it. It has been assigned the value of 12decimal, or 0Chex.

I hope to show in the following text and examples how you can make use of it. But first lets start with its principle of operation. The Magic system is enabled (set into operation) when data is written to a memory location (X) between 0 and 16383dec. or 0 to 3FFFhex. Since the first half of this area is the ROM operating system and the second half is reserved for the ROM game cartridges, we cannot write data to this area. If we try to write to this area, the Magic system knows to add 16384 to our location (X), and instead write to the new location (X+16384) the data modified by the contents of the Magic Register. The type of modification done is determined by the bits that are set in the Magic Register. The bits are assigned as follows:

- |     |   |                                       |
|-----|---|---------------------------------------|
| Bit | 0 | Least significant bit of shift amount |
|     | 1 | Most significant bit of shift amount  |
|     | 2 | Rotate (not supported by our Arcade)  |
|     | 3 | Expand                                |
|     | 4 | OR                                    |
|     | 5 | XOR                                   |
|     | 6 | Flop                                  |
|     | 7 | Not used                              |

The order in which Magic functions are performed is as follows:

Expansion is done first, then Rotation or Shifting, then Flopping, and lastly OR or XOR. As many as four functions can be used at any one time, and any function can be bypassed. Rotate and Shift as well as OR and XOR can not be done at the same time.

Next I will give a quick explanation of the Magic functions. For a more detailed explanation I refer you to the Operating System Manual.\*\* When using the shifter, the data will be shifted 1,2, or 3 pixels to the right, depending on the value of bits 0 and 1 of the Magic Register. Rotate can only be performed by the commercial version of the Arcade. Expand takes the 8 bits of data to be written and expands it to 16 bits by making a 0 bit into a pixel defined by bits 0 and 1 of the expand register, and a 1 bit into a pixel defined by bits 2 and 3 of the expand register. When using the expand option, it is necessary to prepare the expand register (output port 19hex or 25dec) to determine in what way a bit will be expanded into a pixel. This is useful for storing a two-color pattern in one-half the normally required memory space. The OR and XOR functions will first read the memory location (X+16384), then OR or XOR the data that we want to write there with the data that was already there, and then store the result in the location (X+16384). Flop exchanges pixel 0 with pixel 3, and pixel 1 with pixel 2.

Now that we know what the Magic Register can do, let's try an example. Since the Magic Register is a hardware register, it is usable with both machine language and Basic. To make the example simpler, we will do it in Basic.

For ou we will use the XOR function. I will leave the other functions for you to experiment with on your own.

What can we do with XOR? Well, if you have graphics on the screen and you want to place another pattern on the screen without erasing the existing graphics, you can do this by using the XOR function when you write the second pattern to the screen. (Examples of this would be the players moving on the field in baseball, or the galaxians moving over the stars without erasing them.)

By writing the second graphic to the screen again in the same place using the XOR function, we now 'erase' our second pattern and leave the screen like we started. So lets set up our example to see this happen. First we set the screen parameters and put a box on the screen. This is our 'original' pattern. Then we set the Magic Register for XOR (MAGIC = 00100000) which is 20hex or 32dec. Then we will POKE some values to the screen to write our second pattern. (NOTE: using the PRINT, BOX, or LINE command in Basic will clear the Magic Register, this is why we must POKE the second pattern.) Now we see a small pattern in the box and if we POKE the same second pattern to the same place using the XOR function, we will erase our small pattern and leave the box as when we started.

The first program example will POKE, then erase, our small pattern over and over (pressing any key will cause the pattern to appear and disappear as the key is repeatedly pressed.) the second program will jump the pattern across the screen without messing up what was there to start with. Hopefully this tutorial will help you to understand a bit (pun intended) more about what the Magic Register is and what it can do.

Brett Bilbrey 14430 Barclay, Dearborn, MI

```
10 CLEAR
20 &(9)=40
30 FOR A=1006TO 1032;FOR B=1TO 2
40 %(A)=60;%(A+40)=215;%(A+80)=255;%(A+120)=40;%(A+160)=195
50 IF B=1 FOR D=1TO 100;NEXT D
60 NEXT B;NEXT A;B=KP;GOTO 20
>
```

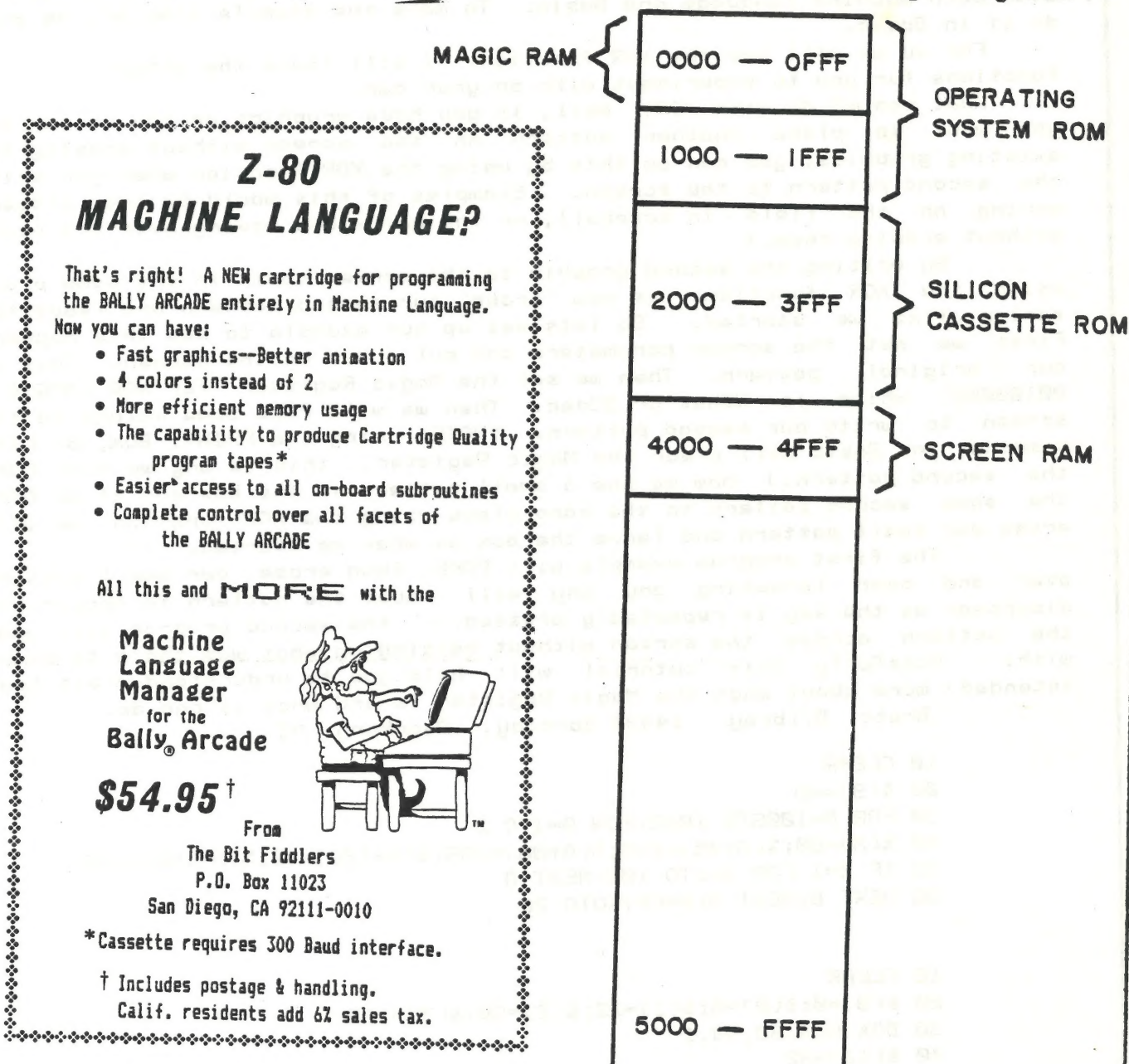
```
10 CLEAR
20 &(9)=0;&(0)=0;&(1)=22;&(2)=66;&(3)=156
30 BOX 0,0,50,50,1
40 &(12)=32
50 A=1016;%(A)=60;%(A+40)=215;%(A+80)=255;%(A+120)=40;%(A+160)=195
60 B=KP;GOTO 40
>
```

---

\*\*This document, also known as The Manual of Hardware and Software, was written by the Bally software design engineering team of Dave Nutting Associates. It covers subroutines, machine operations, and details of the interior workings, plus some specifications of the custom chips. The second half includes disassembled listings for virtually everything. Two points - it talks to the commercial version at times, and just how applicable it is to the current production is not known. Over 300 pages, it is available from the ARCADIAN at \$35, postpaid.



This chart (taken from the Manual referenced on page 25) shows the allocation of memory, and illustrates the location called Magic Register.



W & W Software have temporarily closed down, primarily due to a printer problem, plus the recent move. We'll let you know when they are up and running again.

GREEN SHEET error. Those of you new subscribers who have copies of the green program listings may find a difficult-to-read segment in line 233 of page 4. The second box command should read BOX 34,4,12,9,L

BALLY BACK? The Dorfman article reprinted last time mentioned a Bally return to consumer games - well, so far, they have licensed PacMan to Atari for their machine, and they have licensed Commodore to manufacture some games for the VIC-20.



```

1 . COLUMNS - D. MARTIN
2 NT=0;FC=BC;GOTO 10
3 IF Z=1Z=72
4 IF Z=2Z=68
5 IF Z=3Z=67
6 IF Z=4Z=83
7 RETURN
8 BOX X-3,H,15,9,3;RETURN
9 Z=(Z-1)B7;FOR Y=0TO 6;@(Q+Y)=@(Z+Y);NEXT Y;RETURN
10 CLEAR ;PRINT "      C O L U M N S";PRINT "*** THE MAD PROGRAMMER ***
12 CX=-59;FOR A=1TO 20;TV=42;NEXT A;PRINT
15 CY=0;PRINT "      SHUFFLING CARDS...";@(0)=(RND (8)+1)B10+RND (4);FOR A=1TO 2
0
20 @(A)=(RND (8)+1)B10+RND (4);FOR B=0TO A-1;IF @(B)=@(A)GOTO 20
30 FC=FC+1;NEXT B;BC=BC+8;NEXT A;FC=84;BC=0;R=10
40 FOR T=1TO 3;CLEAR ;CY=30;@(10)=0;FOR A=0TO 6;B=0;C=RM;D=0;E=R
M;F=0;G=RM
50 Z=C;GOSUB 3;C=Z;Z=E;GOSUB 3;E=Z;Z=G;GOSUB 3;G=Z;PRINT #7,B,D,F,;H=30-A*10;X
=-35;CX=X;TV=C;GOSUB 8;X=7;CX=X;TV=E;GOSUB 8;X=49
55 CX=X;TV=G;GOSUB 8;CX=-77;CY=CY-10;@(10)=37+A*20;NEXT A;PRINT " WHICH COLUMN
",;IF T>1PRINT " NOW",
57 PRINT " ?";NT=9;@(10)=176
60 L=KP-48;MU=L+48;IF (L>3)+(L<1)GOTO 60
65 FOR Z=49TO 59;PRINT ;MU=Z;NEXT Z;NT=0;IF L=1M=2;N=3
70 IF L=2M=3;N=1
80 IF L=3M=2;N=1
90 Z=N;Q=21;GOSUB 9;Z=L;Q=28;GOSUB 9;Z=M;Q=35;GOSUB 9;N=20;FOR Z=0TO 6;FOR Q=Z
TO Z+14STEP 7;N=N+1;@(Q)=0;NEXT Q;NEXT Z;NEXT T
100 CLEAR ;PRINT "      YOUR CARD WAS...";Q=0;R=RM;FOR A=16TO 23;@(A)=75;N
EXT A;GOSUB 3;X=0;R=Z
110 CY=0;FOR A=1TO RND (30)+10
115 CX=-5;B=RND (8)+1;Z=RND (4);GOSUB 3;PRINT #1,B,;TV=Z;NEXT A
120 CX=-5;H=8;PRINT #1,Q,;TV=R;GOSUB 8
130 FOR A=23TO 16STEP -1;@(A)=0;NEXT A
140 CY=-24;CX=-65;PRINT "WANT TO SEE IT AGAIN ?";IF KPRUN
>

```

COLUMNS is a computerized version of an old card trick. The computer will randomly select 21 cards (no face cards are used) and shuffle them. They are then laid out in 3 columns containing 7 cards each. You pick any card from the layout, then when the computer asks /WHICH COLUMN/ type in the column which contains your card (1-3 from left to right). The computer will then pick up the cards vertically, by column, and deal them out horizontally, which will rearrange the order. Tell the computer which column your card is in now. The computer will pick them up again vertically and lay them out horizontally, then ask you for the column once more. The computer will tell you the name of your card!

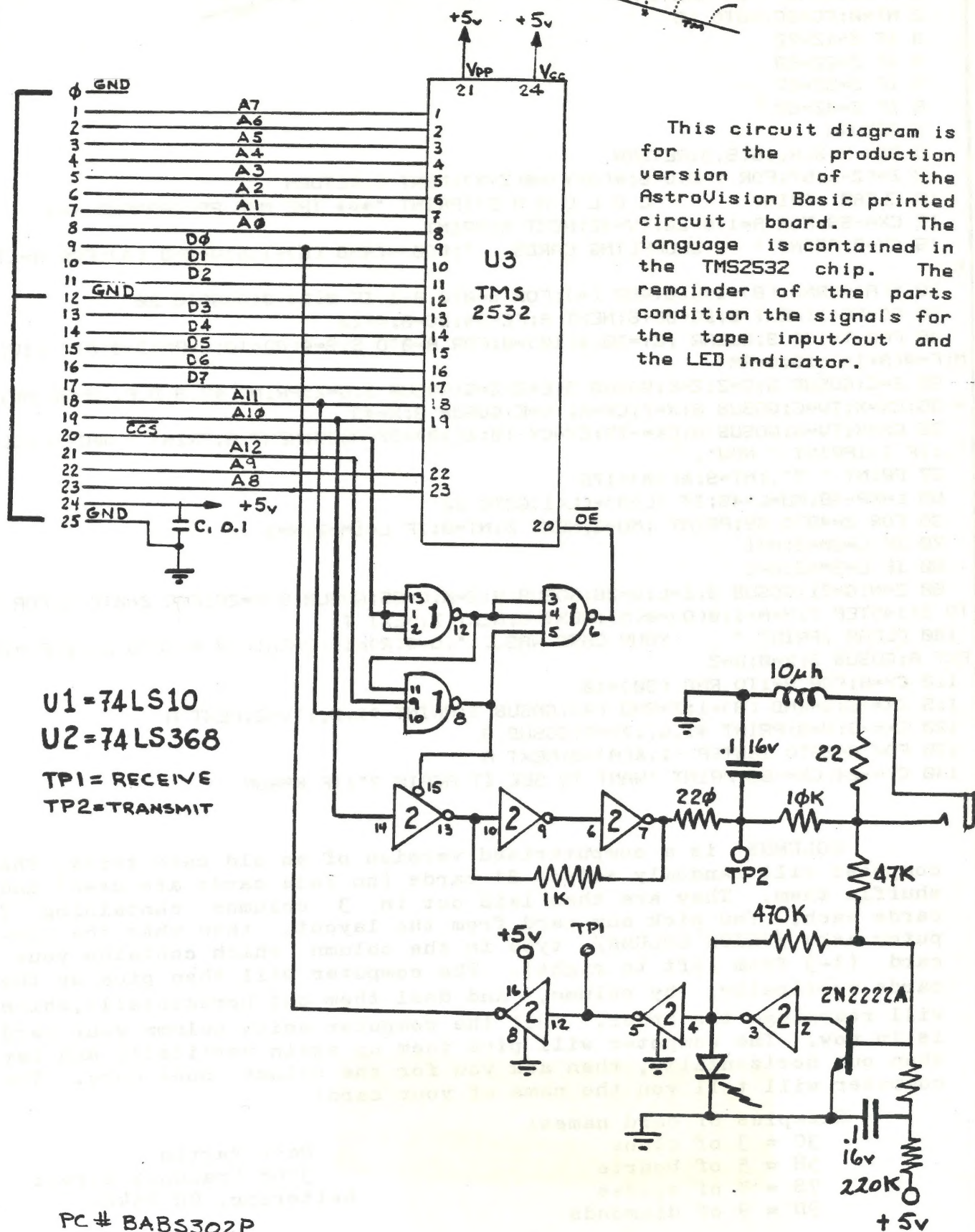
Examples of card names:

3C = 3 of clubs  
 5H = 5 of hearts  
 7S = 7 of spades  
 9D = 9 of diamonds

Dave Martin  
 3408 Braddock Street  
 Kettering, OH 45420



# ARCADIAN





## MACHINE LANGUAGE MANAGER, a review...

We carried an ad for The Bit Fiddlers in the last issue, which offered a new cartridge to be used in the Arcade. I ordered one, and quickly received a package containing the cartridge, a manual, and an overlay card. I asked Al Rathmell to review the product, and his report follows:

"The Machine Language Manager from the Bit Fiddlers is an EPROM plug-in software package for Z-80 machine language program development on the Bally Arcade.

"The user's manual is excellent. There are nine chapters that in a concise and easy-to-understand way, cover the concepts of machine language programming, use of the manager and its utility routines, and some good sample programs.

"The manual also includes a complete source listing of the Machine Language Manager, as well as the Z-80 instruction set in alphabetical order. The users manual is great, and one of the best features of the Machine Language Manager!

"Using the Manager is easy. After reading the users manual once, all that is required is an occasional glance at the quick reference command list on p. 8-1.

"The first obvious difference from using Bally Basic is the small character font. The small font allows up to 15 lines, and 39 characters per line. At first it looks a little strange, especially N and Q, but it is certainly readable and allows about twice as much information to be displayed than Basic. Punching in machine code is simple if you don't make mistakes. Fixing mistakes is a little tedious since the address is shown only every eight bytes. The insert and delete are somewhat difficult to use because you must keep track of the address of the last program opcode.

"The reset function is like the HALT command in Basic. Pushing RESET returns control to the manager without changing memory. Memory is only cleared during power-up.

"The Machine Language Manager is a useful tool for those into machine language programming. Some future additions that would make it better include a search function that would scan a memory range for a specified byte or string of bytes, and an assembly function that would generate opcodes from mnemonic source codes."

The above was written by one who uses machine code programming frequently, not a new-comer. I noted that the text contains some educational material, and some leading-by-the-hand, which is just what I need. As with most computer operations, one needs to "do it yourself" to better understand the operation, and I will be doing so in the future. You will be hearing more about the Machine Language Manager as we utilize this opportunity to study the computer in more detail.

A review of the manual that comes with the cartridge reveals a few interesting details. The first 37 pages contains a mixed bag of general descriptive material, some insights into the Basic language as written by the Bally software team, specific data about the Manager, and instructions as to its operation. There are a few sample programs.

A shift key is included, so that a couple of keys do double duty. These primarily are associated with the use of the tape port (300 baud, requiring the cassette interface), so that the equivalent of Basic commands :PRINT, \*PRINT, :INPUT, and :LIST can be generated.

Following the above, the Bit Fiddlers have included the disassembled Manager program (28 pages), and the Z-80 OpCode in 15 pages.

```

1 CLEAR :J=0;K=0;L=1;M=1;G=1;F=1;NT=0;GOSUB 700
2 CY=20;CX=-40;PRINT " D&D SPEEDWAY
3 PRINT ;CX=-10;PRINT "LAPS
4 PRINT ;CX=-20;PRINT "#1 =";CX=-20;PRINT "#2 =
5 GOSUB 500
9 BOX 0,0,160,88,3;BOX 0,0,150,78,3;BOX 0,0,100,55,3
10 BOX 0,-16,50,17,3
11 GOSUB 800+Rb10
12 &(16)=50;&(17)=154;&(18)=124;&(20)=4;&(22)=222
19 A=58;D=65;B=-27;E=B
20 Z=(KN(1)+128)c20+1;W=(KN(2)+128)c20+1
21 IF JX(1)=1A=A+Z;F=0
22 IF JX(1)=-1A=A-Z;F=0
23 IF JY(1)=1B=B+Z;F=1
24 IF JY(1)=-1B=B-Z;F=1
25 IF JX(2)=1D=D+W;G=0
26 IF JX(2)=-1D=D-W;G=0
27 IF JY(2)=1E=E+W;G=1
28 IF JY(2)=-1E=E-W;G=1
29 S=A;T=B;IF F=0GOSUB 910;GOTO 31
30 GOSUB 900
31 IF (PX(A,B)=0)+(ABS(A)>72)+(ABS(B)>37)BC=90;GOSUB 950;BC=172;GOSUB 105
32 IF A>0IF L=0J=J+1;GOSUB 500;L=1;IF J=QGOTO 960
33 IF A<0IF B>20L=0
34 S=D;T=E;IF G=0GOSUB 910;GOTO 36
35 GOSUB 900
36 IF (PX(D,E)=0)+(ABS(D)>72)+(ABS(E)>37)BC=134;GOSUB 950;BC=172;GOSUB 115
37 IF D>0IF M=0K=K+1;GOSUB 500;M=1;IF K=QGOTO 960
38 IF D<0IF E>20M=0
40 S=A;T=B;IF F=0GOSUB 910;GOTO 42
41 IF F=1GOSUB 900
42 S=D;T=E;IF G=0GOSUB 910;GOTO 50
43 IF G=1GOSUB 900
50 GOTO 20
105 S=A;T=B;IF F=0GOSUB 910;GOTO 107
106 GOSUB 900
107 S=58;T=-27;L=1;GOSUB 900;A=S;B=T;F=1;RETURN
115 S=D;T=E;IF G=0GOSUB 910;GOTO 117
116 GOSUB 900
117 S=67;T=-27;M=1;GOSUB 900;D=S;E=T;G=1;RETURN
500 CY=-12;CX=10;PRINT #1,J;CX=10;PRINT #1,K
510 RETURN
700 PRINT " JOYSTICK STEERS CAR";PRINT " KN ADJUSTS SPEED
710 PRINT ;PRINT " INPUT TRACK";PRINT " (1=EASY TO 5=HARD)";R=KP-48
715 IF (R<1)+(R>5)GOTO 710
720 INPUT " INPUT # OF LAPS"Q;IF QCLEAR ;BC=172;RETURN
810 RETURN
820 S=62;T=0;GOSUB 900;S=-62;GOSUB 900;RETURN
830 GOSUB 820;S=0;T=33;GOSUB 910;T=-33;GOSUB 910;RETURN
840 T=-30;FOR S=-72TO -52STEP 10;T=T+5+RND (10);GOSUB 900;NEXT S;S=0;T=33;GOSUB 910;T=-33;GOSUB 910
841 T=-30;FOR S=52TO 72STEP 10;T=T+5+RND (10);GOSUB 900;NEXT S;RETURN
850 GOSUB 840;S=-50;T=30;GOSUB 910;S=45;GOSUB 910;T=-30;GOSUB 910;S=-50;GOSUB 9

```

D & D SPEEDWAY

Mario DeLaura  
87 Close Hollow Dr.  
Hamlin, NY 14464



# ARCADIAN

```

851 S=-30;T=37;GOSUB 910
852 S=25;T=-37;GOSUB 910;RETURN
900 BOX S,T,2,5,3;BOX S+1,T-1,1,1,3;BOX S-2,T-1,1,1,3;BOX S+1,T+1,1,1,3;BOX S-2
,T+1,1,1,3;RETURN
910 BOX S,T,5,2,3;BOX S-1,T+1,1,1,3;BOX S-1,T-2,1,1,3;BOX S+1,T+1,1,1,3;BOX S+1
,T-2,1,1,3;RETURN
950 FOR P=0TO 50;&(16)=P;NEXT P;&(21)=255;&(23)=255;&(21)=0;&(23)=0;RETURN
960 &(16)=0;&(17)=0;&(18)=0;&(20)=0;&(22)=0
965 CLEAR ;BC=7;CY=0;CX=0;IF J=QPRINT "1
966 IF K=QPRINT "2
970 FOR Y=-30TO 30STEP 20;FOR X=-70TO 70STEP 20;BOX X,Y,10,10,3;NEXT X;NEXT Y
1010 FOR Y=-20TO 20STEP 20;FOR X=-60TO 60STEP 20;BOX X,Y,10,10,3;NEXT X;NEXT Y

```

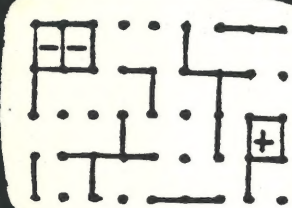
```

10 CLEAR
11 T=0
15 BC=0;FC=12
16 P=0
19 P=P+1
20 FOR A=1TO 81STEP 2
30 BOX 0,0,A,A,3
34 IF FC=32764FC=124
35 FC=FC+8b2
36 IF P=4IF A=77GOTO 290
40 NEXT A
50 IF T<7GOTO 290
55 Q=RND (4)
56 IF Q=1GOTO 350
170 FOR C=-19TO 20
180 D=Cb2
190 BOX C,D,2,2,3
195 BOX D,C,2,2,3
196 IF FC=32764FC=124
197 FC=FC+8b2
200 NEXT C
210 FOR C=-20TO 19
220 D=-(Cb2)
230 BOX D,C,2,2,3
240 BOX C,D,2,2,3
270 NEXT C
280 GOTO 19
290 FOR C=-40TO 40STEP 2
300 BOX 0,C,80,1,1
305 NEXT C
310 FOR C=-40TO 40STEP 2
320 BOX C,0,1,80,1
325 NEXT C
327 P=P+1
328 IF P=7GOTO 20
330 T=T+1
335 IF T<5GOTO 350
340 GOTO 16
350 FOR C=1TO 80STEP 8
360 BOX 0,0,C,C,3
370 NEXT C;GOTO 20

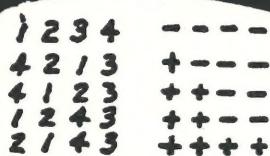
```

The above interesting graphics program is by Ken Springsteen,  
1935 Winding Hill Rd.#1008 Davenport, IA 52807

## EDGE SOFTWARE



1. DOTS



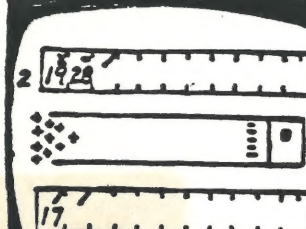
2. MASTERMIND

TAPE # 2 \$12.95

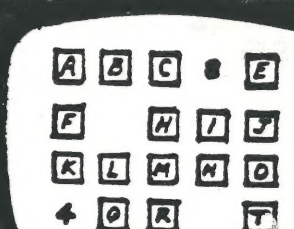
ORDER EITHER TAPE.

SEND CHECK OR MONEY ORDER TO :

## EDGE SOFTWARE



1. BOWL



2. MATCH

TAPE #1 \$12.95

EDGE  
12046 FLAMBEAU DR.  
PALOS HTS, ILLINOIS 60463



**\*\*AD**

SFP has three original Bally Basic in/orig. carton at \$24.95 each. Also have all Bally/Astrovision Videocades (that are released) in stock, at a discount to ARCADIAN readers. Revolving Cassette Rack at \$5.95 holds 20 Videocades in cases, or 32 cassettes without cases. Reg. Price \$7.95 . CA add 6% tax. For Free price list; SFP D-A, 1064 N. Alta, Dinuba, CA. 93618 (209) 591-0555.

32

**ARCADIAN**

Robert Fabris, Holiday Well-wisher  
3626 Morrie Drive  
San Jose, CA 95127-9990

The SOURCE  
TCD 959

**FIRST CLASS**  
**U. S. POSTAGE**  
**PAID**  
Sunnyvale, CA  
Permit No. 931

63301BRG2917H  
ARTHUR BERGANTZ  
2917 HEADLAND DRIVE  
ST CHARLES, MO

63301 \*